ABSTRACT

Cross-Site Request Forgery(CSRF) is a widely exploited web site vulnerability. In a CSRF attack, a malicious site instructs a victim's browser to send a request to an honest site, as if the request were part of the victim's interaction with the honest site, leveraging the victim's network connectivity and the browser's state, to disrupt the integrity of the victim's session with the honest site.

In this work we prepare a web-application, "Online Trusted Bank", to demonstrate a CSRF attack and then study two existing defenses against the attack. The first approach [12][13][14] we used to defend CSRF attacks involve sending additional information i.e. token in each HTTP request. If a request is missing a validation token or the token does not match the expected value, the server rejects the request. The second approach [15] uses CAPTCHA to detect the attack. CAPTCHAs are used to prevent automated scripts from submitting forms on a website. Since the attackers cannot read the CAPTCHA image, it became difficult for them to determine the actual text.

Then we suggest a graph based algorithm to defend the conducted CSRF attack on our developed "Online trusted Bank" application. The graph based method uses a sub-graph matching between a graph prepared from a legitimate web profile containing white list and a graph prepared based on run-time web document of the authenticated user who is currently logged into the "Online Trusted Bank" application. Experimental evaluation is performed to identify deviation of a run-time web document using web profile and it is computed based on web-graph[17].

Finally, we provide a comparative study of the defence mechanisms that we have used to detect the CSRF attack against "Online trusted Bank" application, listing the advantages and disadvantages of each mechanism. The study shows that the proposed graph based method has a few advantages over the other two implemented methods.